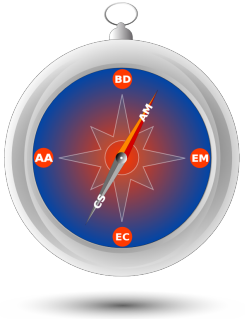


Beam Dynamics Frameworks

James Amundson
Fermilab

with
Panagiotis Spentzouris *Fermilab*,
Douglas Dechow *Tech-X*

Overview

- Why we are here
- A (pre-)historical view of SciDAC Beam Dynamics frameworks
- What we need to move forward
- Some questions

Disclaimers

- Mostly a talk about the future
- Much discussion of ML/I and Synergia2
  - Only examples
  - One of the main points is to mix together more pieces of software

Why we are here

- Our proposal was accepted
  - From the proposal's Project Summary:

The SciDAC1 accelerator project, a partnership of accelerator computationalists, applied mathematicians, and computer scientists, generated a suite of parallel accelerator simulation tools. These were applied to important accelerator projects of the DOE. Under SciDAC2, these tools will be enhanced to contain new capabilities as needed by HEP projects, such as the ILC, the LHC, the Tevatron, and PEP-II, and for Advanced Acceleration research; NP projects, such as CEBAF and RHIC, the CEBAF and RHIC upgrades, RIA, and an NP electron collider, including ELIC and eRHIC; and BES projects, such as LCLS, NSLS-II, SNS, and upgrades to the APS.

This simulation suite will contain a comprehensive set of interoperable components for beam dynamics, electromagnetics, electron cooling, and advanced accelerator modeling.

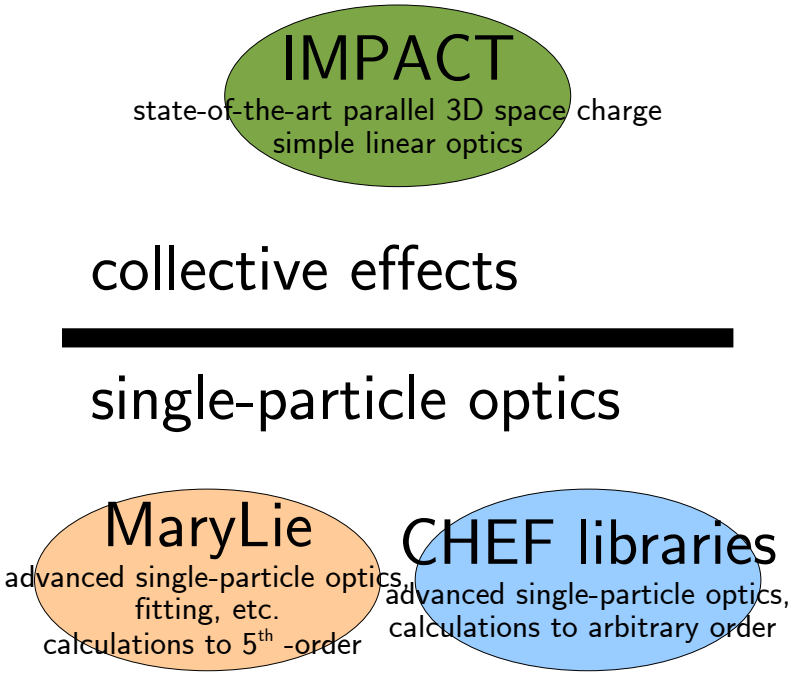
Why we are here, cont.

- From the proposal's Executive Summary:

Under SciDAC2--recognizing the complexity, precision, and beam intensity requirements of next generation accelerators--our paradigm will change from single machine, single-component simulations to end-to-end (multi-stage or complete system), multi-physics simulations. Building upon the foundation laid under SciDAC1, we will extend our terascale capabilities to the petascale, and add new capabilities to deliver a comprehensive, fully integrated accelerator simulation environment.

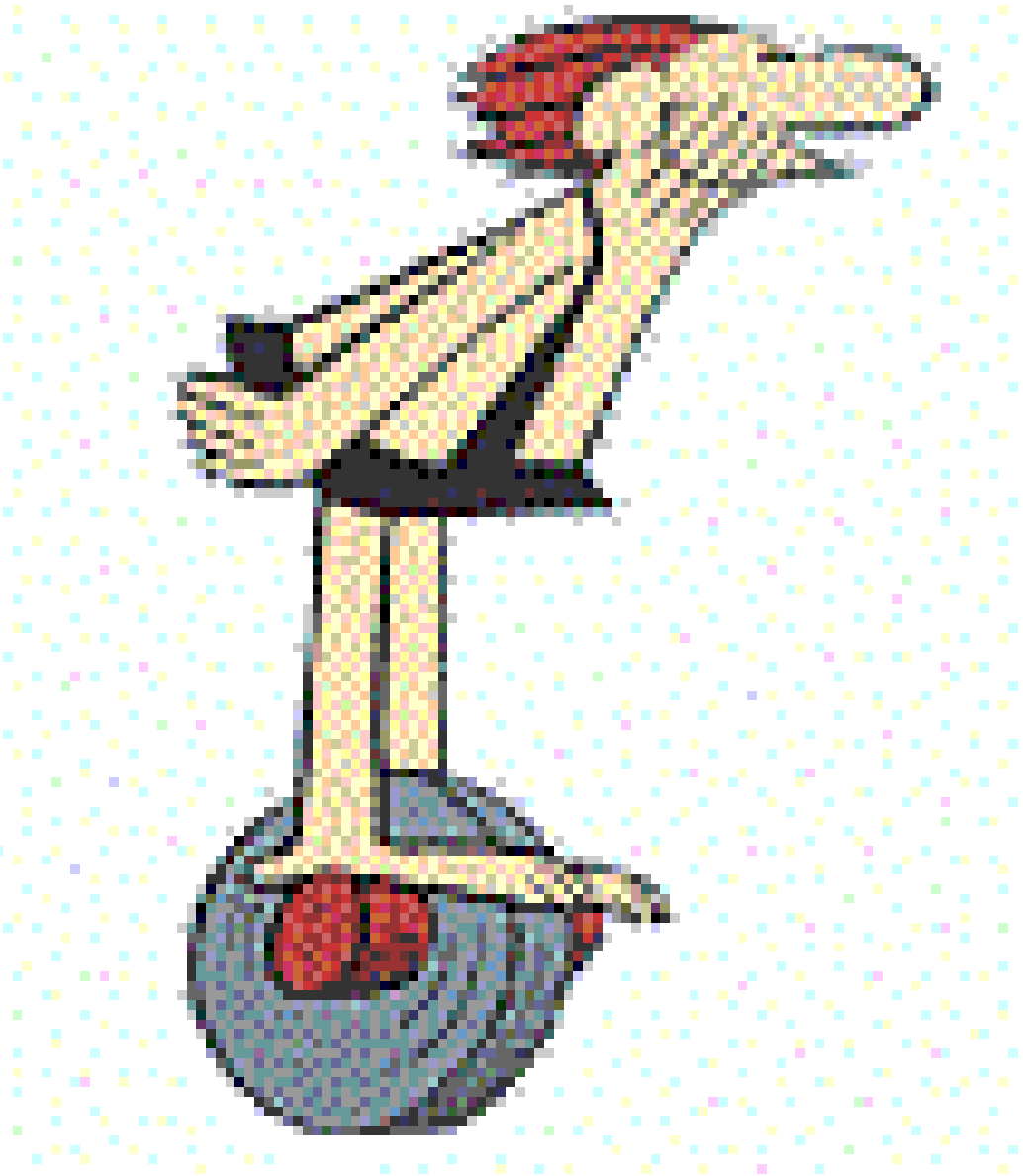
BD Frameworks before SciDAC1

- Self-contained applications



- The Stone Age
  - Individuals/families in separate caves
    - Limited collaboration
  - Straightforward, effective tools
    - Highly limited interoperability

Summary of pre-SciDAC1 era framework development



BD Framework development during SciDAC1

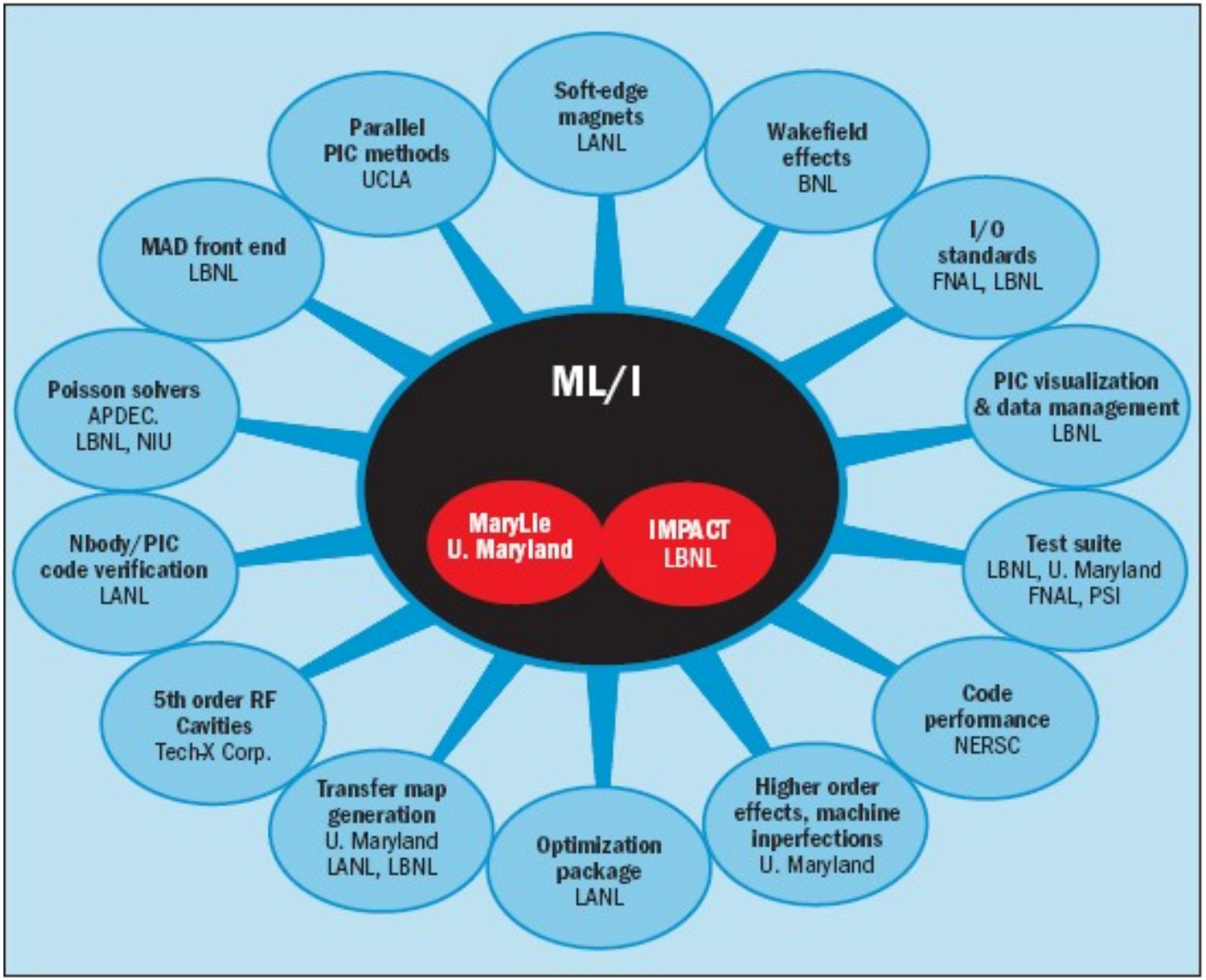
- MaryLie + IMPACT
  - MaryLie/IMPACT
- CHEF + IMPACT
  - Synergia

- The Bronze Age
  - Emergence of farmsteads and ethnic groups
    - Pooling of resources and skills
  - First tools formed from an alloy
    - Copper + Tin
    - Much greater sophistication possible

Summary of SciDAC1 era framework development

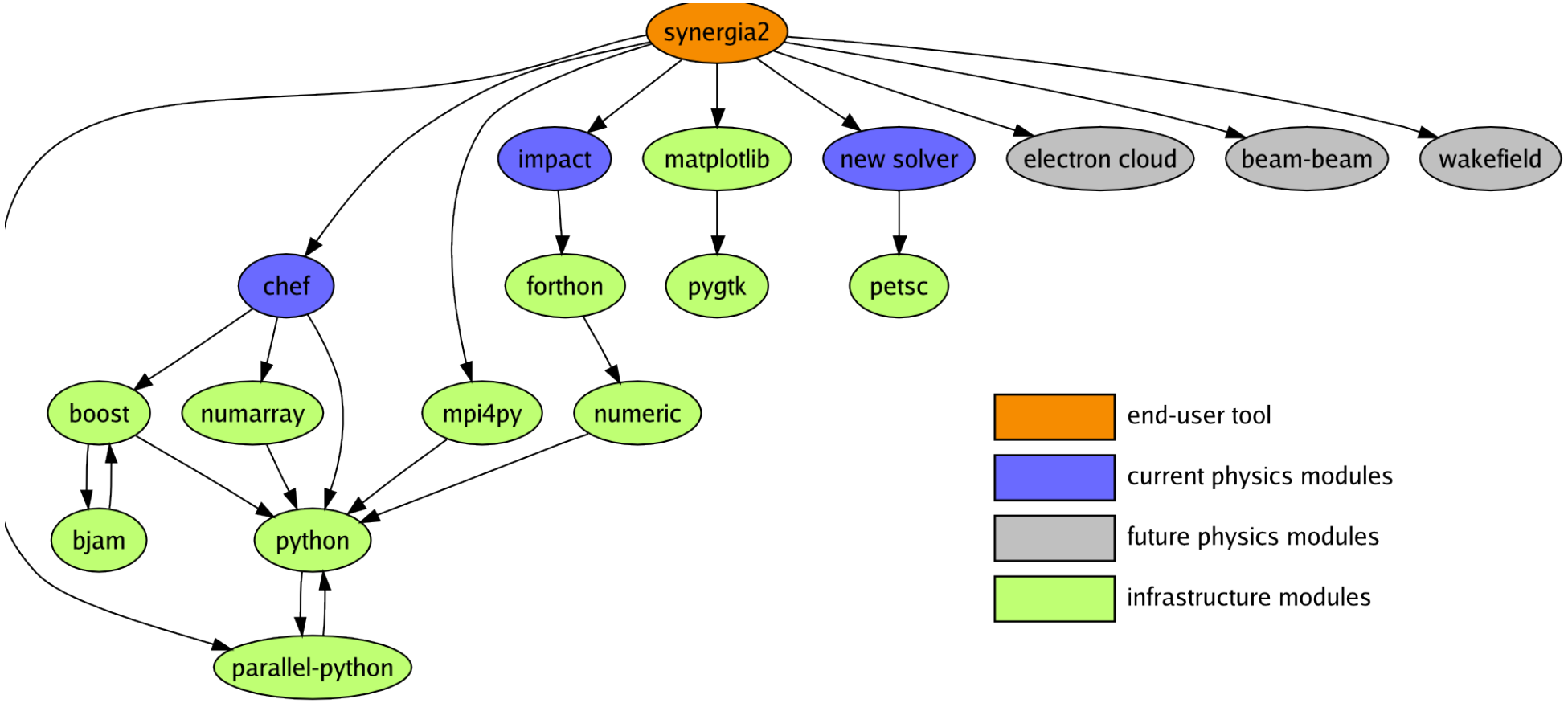
SciDAC1 accomplishments

ML/I



More SciDAC1 accomplishments

Synergia -> Synergia2



Why we need to move to the next Age

- Progress during SciDAC1 was hard work
  - Integration was accomplished one piece at a time
    - Work not immediately applicable to any other application – many redundancies
    - Standalone testing not possible
  - Inter-language issues a recurring problem
    - Mechanics of calling one language from another
      - Includes difficulties in cross-platform compilation of mixed language code
    - Difficulties in defining interfaces across languages

Reasons to move on, cont.

- Efficient parallel performance on next-generation machines will require algorithmic work
  - Future platforms
    - Capability machines (supercomputers)
      - (even more) massively parallel architectures
        - Specialized communications patterns
    - Commodity machines (clusters and desktops)
      - Multi-core is on the way
    - See, e.g., http://cscads.rice.edu/workshops/july2007/lib-slides
- New algorithms may require structural changes in our software
  - Requires flexible code

Reason to move on, in summary

- The framework techniques employed to date in ML/I and Synergia2 scale poorly for the requirements of SciDAC2.
- We need to get past thinking about our software as individual programs and start thinking about them as pieces of "***a comprehensive, fully integrated accelerator simulation environment.***"

SciDAC2 requires a move to the Iron Age

- The Iron Age
  - Formation of cities and states
    - True collaboration on the details of daily life
    - Strength in numbers
  - Strong, complex tools built from modern alloys
    - Iron
      - Even steel!
      - Unlike bronze, can be sharpened without reforging
    - Requires input from outside the farmstead

The SciDAC2 era of framework development

What we need to do to move ahead

- In general
  - We physicists need to take advantage of the tools and expertise provided by the Applied Math/Computer Science portion of our project (and greater community)
    - Algorithmic
    - Infrastructure
  - We all need to work harder on collaborative software
    - Design
    - Infrastructure

Infrastructure from AM/CS

- Component-based architecture
  - Specifically as defined by The Common Component Architecture Forum, http://www.cca-forum.org/
    - Our project is a major customer
  - Inter-language issues solved for us
  - Interface definition *mechanics* solved for us
  - Interface definition itself is up to us
    - We can expect help from the CS professionals, however
  - See “Components for Beam Dynamics” talk on Tuesday

Component Advantages

- True interoperability
  - Really advantages 1-10
- Eases incorporation of new members of collaboration
  - Components will not need to be re-adapted to every application
- Better testing
  - Possible to perform tests decoupled from parent framework
- Better incorporation of algorithmic improvements
  - Especially from outside contributors
    - Only need to understand component, not entire framework

Component Challenges

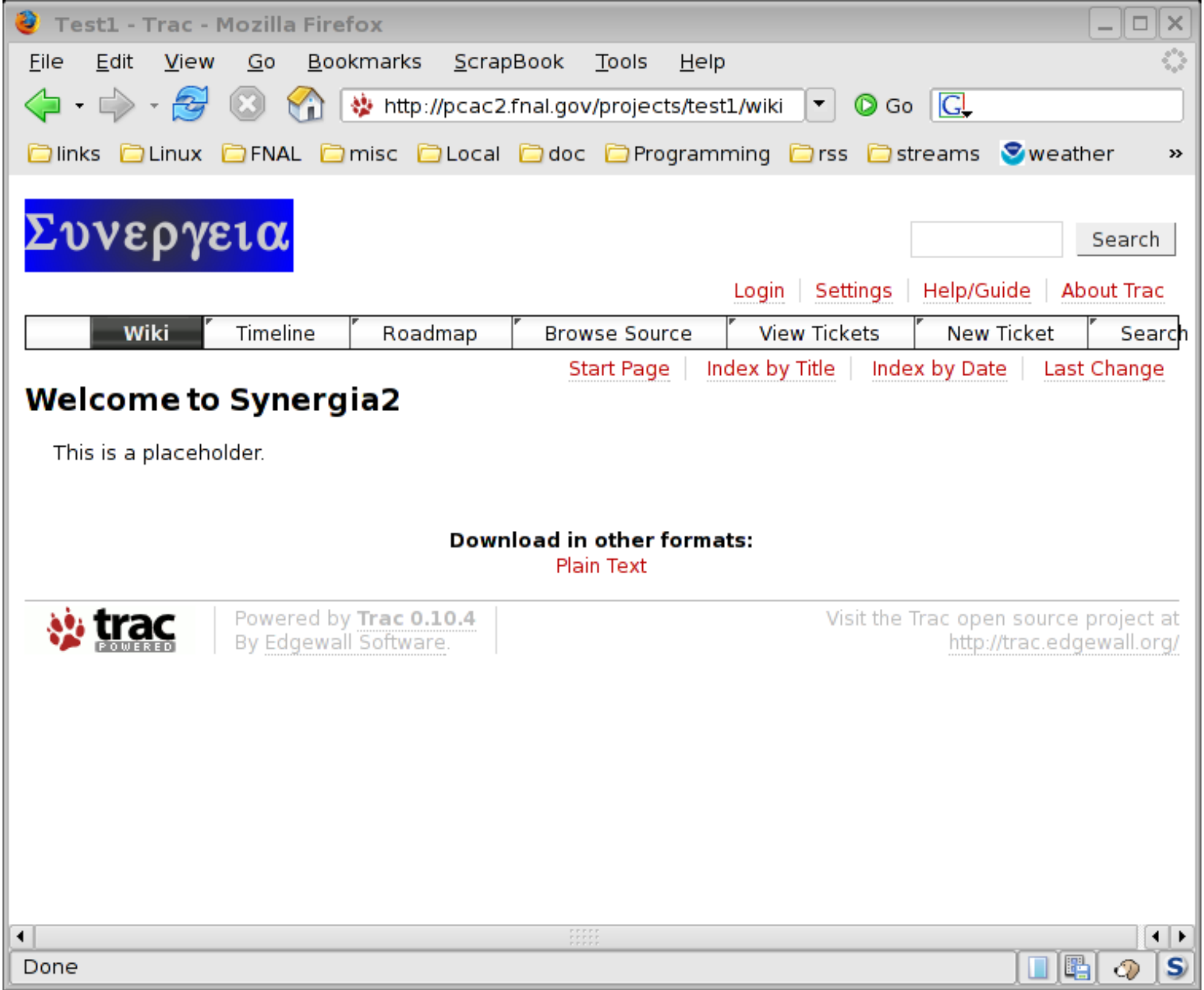
- Getting started
  - That's what the (CS) professionals are for
  - Prototype work with Synergia2 and ML/I
    - The *point* is to not be specific to Synergia2 or ML/I
- Defining the interfaces
  - Really challenges 2-99
  - Building upon experience with Synergia2 and ML/I
    - Again the *point*...
  - Real collaboration necessary
    - Start in breakout sessions

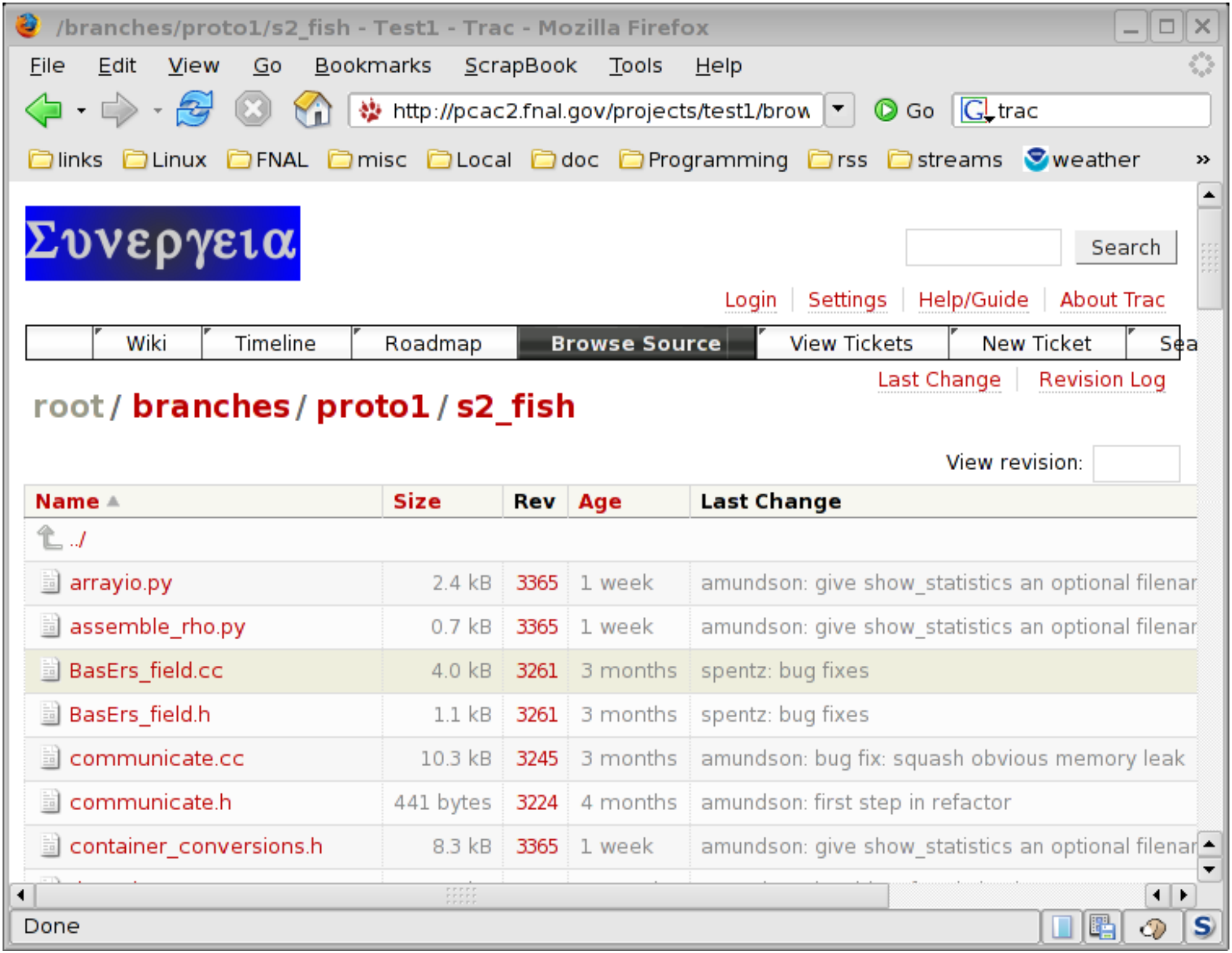
Questions: Collaboration Infrastructure

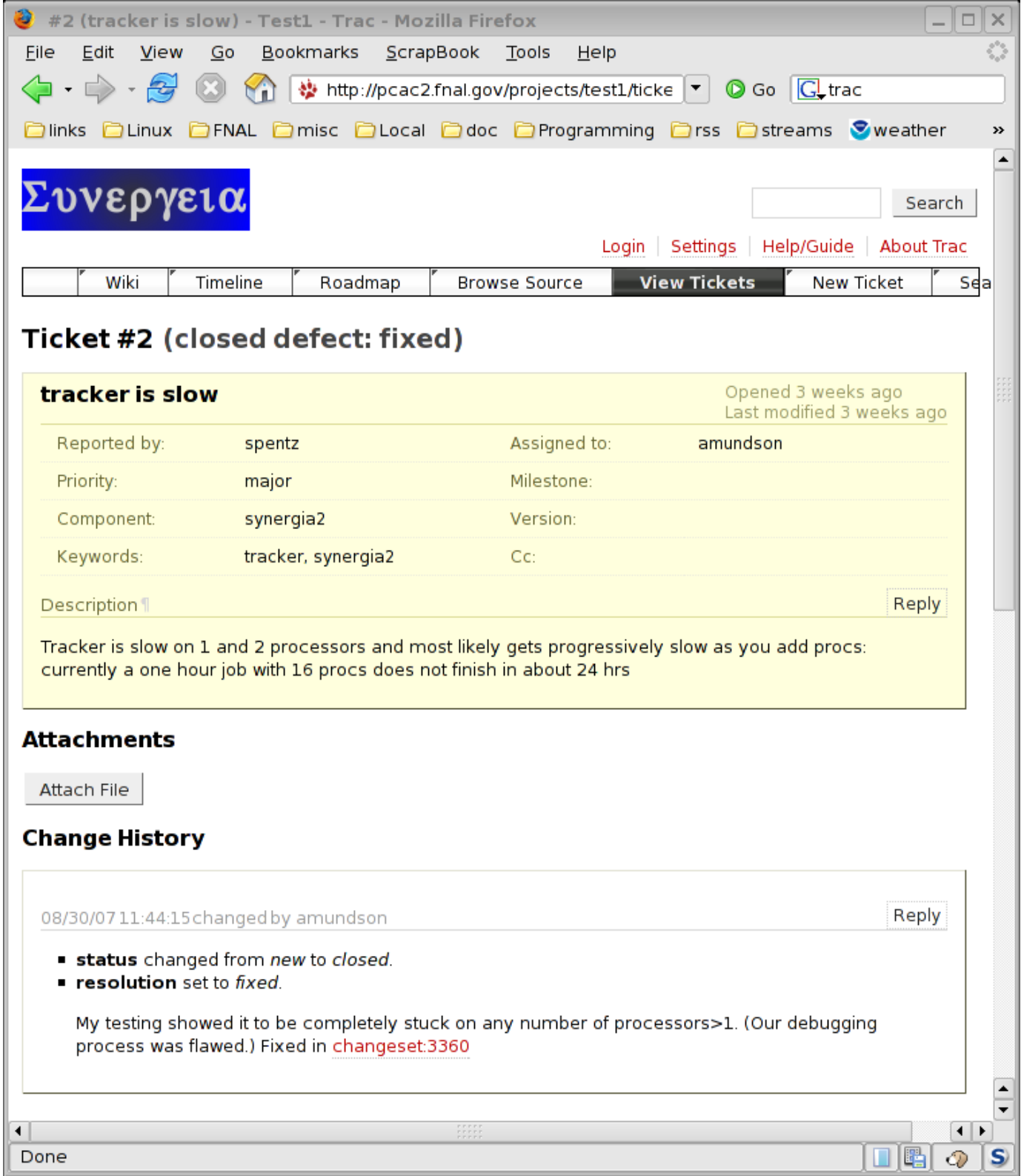
- Now is the time to setup the infrastructure for collaboration within the project
- One possibility: project-wide hosting through a Trac Server

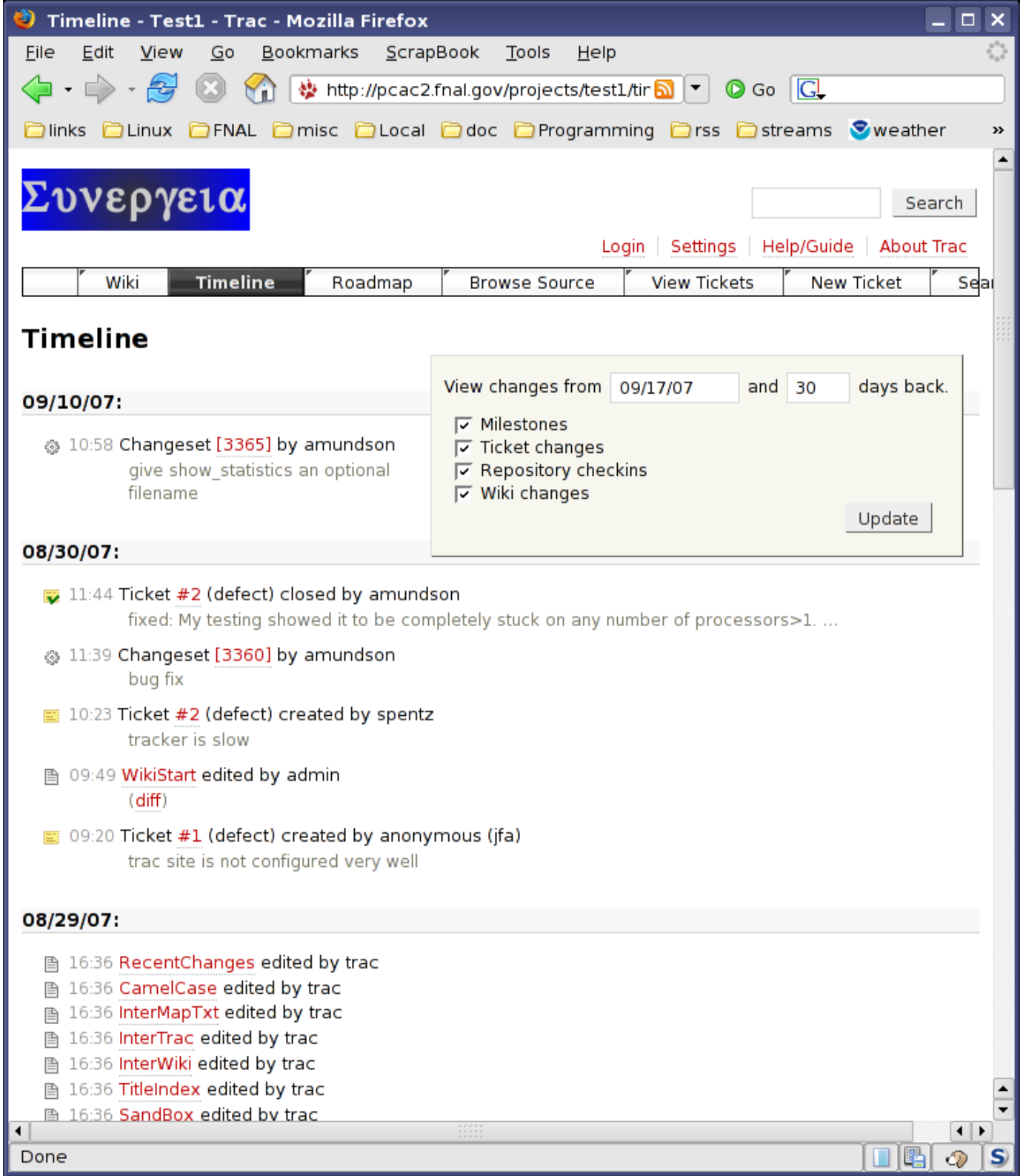
Trac

- http://trac.edgewall.org/
  - Subversion server and interface
  - Bug tracking
  - Wiki-based content
  - Access control
  - Many community-supported plugins, etc.
    - Automated build/testing


Test1 - Trac - Mozilla Firefox
File Edit View Go Bookmarks ScrapBook Tools Help
http://pcac2.fnal.gov/projects/test1/wiki
Go
links Linux FNAL misc Local doc Programming rss streams weather
Συνεργεια
Search
Login Settings Help/Guide About Trac
Wiki Timeline Roadmap Browse Source View Tickets New Ticket Search
Start Page Index by Title Index by Date Last Change
Welcome to Synergia2
This is a placeholder.
Download in other formats:
Plain Text
trac POWERED
Powered by Trac 0.10.4
By Edgewall Software.
Visit the Trac open source project at
http://trac.edgewall.org/
Done


/branches/proto1/s2_fish - Test1 - Trac - Mozilla Firefox
File Edit View Go Bookmarks ScrapBook Tools Help
http://pcac2.fnal.gov/projects/test1/brow
Go
trac
links Linux FNAL misc Local doc Programming rss streams weather
Συνεργεια
Search
Login Settings Help/Guide About Trac
Wiki Timeline Roadmap Browse Source View Tickets New Ticket Sea
Last Change Revision Log
root / branches / proto1 / s2_fish
View revision:
Name Size Rev Age Last Change
../
arrayio.py 2.4 kB 3365 1 week amundson: give show_statistics an optional filenar
assemble_rho.py 0.7 kB 3365 1 week amundson: give show_statistics an optional filenar
BasErs_field.cc 4.0 kB 3261 3 months spentz: bug fixes
BasErs_field.h 1.1 kB 3261 3 months spentz: bug fixes
communicate.cc 10.3 kB 3245 3 months amundson: bug fix: squash obvious memory leak
communicate.h 441 bytes 3224 4 months amundson: first step in refactor
container_conversions.h 8.3 kB 3365 1 week amundson: give show_statistics an optional filenar
Done


#2 (tracker is slow) - Test1 - Trac - Mozilla Firefox
File Edit View Go Bookmarks ScrapBook Tools Help
http://pcac2.fnal.gov/projects/test1/ticke
Go
trac
links Linux FNAL misc Local doc Programming rss streams weather
Συνεργεια
Search
Login Settings Help/Guide About Trac
Wiki Timeline Roadmap Browse Source View Tickets New Ticket Sea
Ticket #2 (closed defect: fixed)
tracker is slow
Opened 3 weeks ago
Last modified 3 weeks ago
Reported by: spentz
Assigned to: amundson
Priority: major
Milestone:
Component: synergia2
Version:
Keywords: tracker, synergia2
Cc:
Description ¶
Reply
Tracker is slow on 1 and 2 processors and most likely gets progressively slow as you add procs: currently a one hour job with 16 procs does not finish in about 24 hrs
Attachments
Attach File
Change History
08/30/07 11:44:15 changed by amundson
Reply
status changed from new to closed.
resolution set to fixed.
My testing showed it to be completely stuck on any number of processors>1. (Our debugging process was flawed.) Fixed in changeset:3360
Done


Timeline - Test1 - Trac - Mozilla Firefox
File Edit View Go Bookmarks ScrapBook Tools Help
http://pcac2.fnal.gov/projects/test1/tir
Go
links Linux FNAL misc Local doc Programming rss streams weather
Συνεργεια
Search
Login | Settings | Help/Guide | About Trac
Wiki Timeline Roadmap Browse Source View Tickets New Ticket
Timeline
View changes from 09/17/07 and 30 days back.
Milestones
Ticket changes
Repository checkins
Wiki changes
Update
09/10/07:
10:58 Changeset [3365] by amundson
give show_statistics an optional filename
08/30/07:
11:44 Ticket #2 (defect) closed by amundson
fixed: My testing showed it to be completely stuck on any number of processors>1. ...
11:39 Changeset [3360] by amundson
bug fix
10:23 Ticket #2 (defect) created by spentz
tracker is slow
09:49 WikiStart edited by admin
(diff)
09:20 Ticket #1 (defect) created by anonymous (jfa)
trac site is not configured very well
08/29/07:
16:36 RecentChanges edited by trac
16:36 CamelCase edited by trac
16:36 InterMapTxt edited by trac
16:36 InterTrac edited by trac
16:36 InterWiki edited by trac
16:36 TitleIndex edited by trac
16:36 SandBox edited by trac
Done

A Trac server at FNAL

- FNAL CD very experienced in supporting infrastructure for large, distributed collaborations
  - CDF, D0, US CMS, etc.
- Question remains: If we build it, will you come?
  - We have already found Trac useful enough that we will use it internally, anyway
- A discussion topic for breakout sessions

More questions

- What should we do about software distribution?
  - Part of our mandate
  - Must be compilable by someone other than the authors
    - Component architecture inevitably increases build complexity
      - Contractor helps with Synergia2, now used by CCA
  - Trac server would be a logical focal point
  - Another topic for breakout

Summary

- We have ambitious goals for SciDAC2
  - Multi-physics
  - Effective use of next-generation hardware
- Component architecture necessary for both goals
  - Older model requires too much work
- Collaborative issues for breakout sessions
  - Project-wide Trac server?
  - Software distribution?
  - Component interfaces